

# The New Multi-Edge Metric-Constrained PEG/QC-PEG Algorithms for Designing the Binary LDPC Codes With Better Cycle-Structures

Xuan He, *Student Member, IEEE*, Liang Zhou, and Junyi Du, *Student  
Member, IEEE*

## Abstract

To obtain a better cycle-structure is still a challenge for the low-density parity-check (LDPC) code design. This paper formulates two metrics firstly so that the progressive edge-growth (PEG) algorithm and the approximate cycle extrinsic message degree (ACE) constrained PEG algorithm are unified into one integrated algorithm, called the metric-constrained PEG algorithm (M-PEGA). Then, as an improvement for the M-PEGA, the multi-edge metric-constrained PEG algorithm (MM-PEGA) is proposed based on two new concepts, the multi-edge local girth and the edge-trials. The MM-PEGA with the edge-trials, say a positive integer  $r$ , is called the  $r$ -edge M-PEGA, which constructs each edge of the non-quasi-cyclic (non-QC) LDPC code graph through selecting a check node whose  $r$ -edge local girth is optimal. In addition, to design the QC-LDPC codes with any predefined valid design parameters, as well as to detect and even to avoid generating the undetectable cycles in the QC-LDPC codes designed by the QC-PEG algorithm, the multi-edge metric constrained QC-PEG algorithm (MM-QC-PEGA) is proposed lastly. It is verified by the simulation results that increasing the edge-trials of the MM-PEGA/MM-QC-PEGA is expected to have a positive effect on the cycle-structures and the error performances of the LDPC codes designed by the MM-PEGA/MM-QC-PEGA.

## Index Terms

Approximate cycle extrinsic edge degree (ACE), girth, low-density parity-check (LDPC) code, progressive edge-growth (PEG), quasi-cyclic (QC).

Part of this work [1] has been published in the 2015 IEEE International Symposium on Information Theory.

## I. INTRODUCTION

The error floor performance of the low-density parity-check (LDPC) code [2] over the additive white Gaussian noise (AWGN) channel is closely related to a cycle-structure called the trapping set (TS) [3]. However, it was shown in [4] that to find or even to approximate the minimum size of the TS in a Tanner graph (TG) [5] is NP-hard. In addition, the methods in [6]–[9], which try to directly optimize the elementary trapping sets (ETs) [3] real timely during the construction process of the LDPC codes, hold a high realization complexity as well as a high computational complexity, and sometimes will fail the construction. With regard to this, some other widely used cycle-structures, such as the girth [10], the approximate cycle extrinsic message degree (ACE) [11], [12], and the ACE spectrum [13]–[16], which can be calculated simply and efficiently, have thus been used in this paper for the design and the analysis of the LDPC code.

Since good cycle-structures and good error performances have been achieved by the progressive edge-growth (PEG) algorithm [10] as well as the ACE constrained PEG algorithm [14], while both of the algorithms hold a low realization complexity and a polynomial computational complexity, and will never fail the construction, thus our work is closely related to the ideas in [10] and [14]. To be specific, this paper formulates two metrics firstly so that the PEG algorithm [10] and the ACE constrained PEG algorithm [14] are unified into one integrated algorithm, called the metric-constrained PEG algorithm (M-PEGA). Then, as an improvement for the M-PEGA, the *multi-edge metric-constrained PEG algorithm (MM-PEGA)* is proposed based on two new concepts, the *multi-edge local girth* and the *edge-trials*. The MM-PEGA with the edge-trials, say a positive integer  $r$ , is called the  $r$ -edge M-PEGA, which is implemented under the framework of the M-PEGA but adopts a different *selection strategy*. More precisely, the  $r$ -edge M-PEGA constructs each edge of the non-quasi-cyclic (non-QC) LDPC code graph through selecting a check node (CN) whose  $r$ -edge local girth is optimal, instead of through selecting a CN whose one-edge local girth is optimal as that in the M-PEGA. It's illustrated that the one-edge M-PEGA is equivalent to the M-PEGA. In addition, to calculate the multi-edge local girth, a depth-first search (DFS) [17] like algorithm is proposed.

QC-LDPC codes are more hardware-friendly compared to other types of LDPC codes in encoding and decoding. Encoding of the QC-LDPC codes can be efficiently implemented using simple shift registers [18]. In addition, the revolving iterative decoding algorithm [19], [20] significantly reduces the hardware implementation complexity of a QC-LDPC decoder. So,

amount of researchers show great interest in the construction of the QC-LDPC codes. At the same time, well designed QC-LDPC codes perform as well as other types of LDPC codes [7], [9], [16], [20]–[34].

In [20]–[22], construction methods for the QC-LDPC code based on the finite field were proposed. These methods usually firstly construct the complete check matrix of the QC-LDPC code, in which each circulant is a *circulant permutation matrix (CPM)*, and then adopt the masking technique [35] to adjust the degree distributions of both the variable nodes (VNs) and the CNs. In [23]–[25], the Chinese Remainder Theory (CRT) is adopted to accelerate the construction process of the QC-LDPC code, keeping the girth of the target LDPC code not smaller than that of the base matrix. Methods in [7], [16], [26]–[31] construct the QC-LDPC code by carefully cyclically lifting the protograph. In order to achieve the desirable large girth, the constraints which have been derived in [32] to ensure the corresponding girth must be satisfied. However, it's usually impossible to satisfy the constraints unless the protograph is very sparse and the lifting size is quite large. Moreover, in [7], [16], [20]–[32], the design parameters, such as the size of the check matrix, the degree distribution of the check matrix, and the size of the circulant, are not as flexible as that in the QC-PEG algorithm (QC-PEGA) [33]. Instead, the QC-PEGA [33] is suitable for designing the QC-LDPC code with any predefined valid design parameters. However, the QC-PEGA [33] sometimes results in 4-cycles (cycles of length 4) just in a single circulant of the check matrix when there contain multiple edges. To avoid generating 4-cycles in a single circulant of the check matrix, the circulant-permutation-PEG algorithm (CP-PEGA) [34] restricts each nonzero circulant of the check matrix to be a CPM during the construction process of the QC-PEGA [33].

To flexibly design the QC-LDPC codes with better cycle-structures, an improvement for the QC-PEGA [34], called the *multi-edge metric-constrained QC-PEG algorithm (MM-QC-PEGA)*, is proposed in this paper. On the one hand, the MM-QC-PEGA is implemented under the framework of the QC-PEGA [33] so that it can construct the QC-LDPC codes with any predefined valid design parameters. On the other hand, following the idea of the  $r$ -edge M-PEGA, the  $r$ -edge M-QC-PEGA constructs each edge of the QC-LDPC code graph through selecting a CN whose  $r$ -edge local girth is optimal. As a result, the undetectable cycles in the QC-LDPC codes designed by the QC-PEGA [33] and the CP-PEGA [34] become detectable and even avoidable in the codes designed by the MM-QC-PEGA.

To investigate the cycle-structure as well as the error performance, plenty of simulations have

been performed over the binary LDPC codes, which are designed by the MM-PEGA, the MM-QC-PEGA, and some of the conventional LDPC code design algorithms respectively. According to the simulation results, the proposed algorithms, i.e., the MM-PEGA and the MM-QC-PEGA, are more effective than the conventional design algorithms in terms of designing the LDPC codes with better cycle-structures and better error performances. In addition, compared to the non-QC-LDPC codes designed by the MM-PEGA, the QC-LDPC codes, which are designed by the MM-QC-PEGA with the similar design parameters, sometimes achieve better cycle-structures and better error performances.

The rest of this paper is organized as follows. Section II introduces some preliminaries, notations, and backgrounds. Section III firstly defines the multi-edge local girths and the edge-trials, following which the MM-PEGA is proposed. Then, a DFS like algorithm is proposed to calculate the multi-edge local girths. Section IV proposes the MM-QC-PEGA to design the QC-LDPC codes. Section V presents some simulation results. And finally, Section VI concludes the paper.

## II. PRELIMINARIES, NOTATIONS, AND BACKGROUNDS

### A. Graph

A graph is denoted as  $G = (V, E)$ , with  $V$  the set of nodes and  $E$  the set of edges. An edge connecting nodes  $u_0$  and  $u_1$  is denoted as  $(u_0, u_1)$ . At the same time,  $(u_0, u_1)$  is called incident to  $u_0$  and/or  $u_1$ , as well as is regarded as an (incident) edge of  $u_0$  and/or  $u_1$ . A path with length- $L$  connecting nodes  $u_0$  and  $u_L$  is denoted as  $u_0 u_1 \cdots u_L$ , where  $u_i \neq u_j$  and  $u_j \neq u_t$  for  $0 \leq i < j < t \leq L$ . Specially, when  $u_0 = u_L$ , it forms a length- $L$  (size- $L$ ) cycle.  $\forall x, y \in V$ , if there exists at least one path connecting them,  $x$  and  $y$  are said to be connected. In such case, the distance between  $x$  and  $y$  is defined as the length of the shortest path connecting them. However, when  $x$  and  $y$  are not connected, their distance is defined as  $\infty$ .

### B. Binary LDPC Code and Its Tanner Graph

A binary LDPC code can be represented by its check matrix  $\mathbf{H} = [h_{i,j}]_{m \times n}$ , where  $h_{i,j} \in GF(2)$  for  $0 \leq i < m$  and  $0 \leq j < n$ . Also, it can be represented by its TG  $G = (V_c \cup V_v, E)$ , where  $V_c = \{c_i | 0 \leq i < m\}$  is the set of the CNs while  $c_i$  is the  $i$ -th CN of the TG which corresponds to the  $i$ -th row of  $\mathbf{H}$ , and  $V_v = \{v_j | 0 \leq j < n\}$  is the set of the VNs while  $v_j$  is the  $j$ -th VN of the TG which corresponds to the  $j$ -th column of  $\mathbf{H}$ , and  $(c_i, v_j) \in E$  if and only if

$h_{i,j} = 1$ . In this paper, it makes no difference when referring to  $\mathbf{H}$  and/or its corresponding TG  $G$ . In addition, only  $E$  may denote a different set of edges during the construction process of the TG while  $V = V_c \cup V_v$  keeps invariant. Furthermore, denote  $E_{v_j} = \{(c_i, v_j) | 0 \leq i < m\}$  for  $0 \leq j < n$  as the set including all the incident edges of the VN  $v_j$ , and denote  $D = \{d_{v_j} | 0 \leq j < n\}$  as the set including the degrees of all VNs, where  $d_{v_j}$  is the degree of the VN  $v_j$ .

### C. ACE and ACE Spectrum

In the TG, the ACE [11] value of a path is defined as  $\sum_j (d_{v_j} - 2)$ , where  $d_{v_j}$  is the degree of the  $j$ -th VN of the path, and the summation is taken over all VNs of the path. The minimum path ACE metric [14] between two arbitrary nodes  $x$  and  $y$  is defined as the minimum ACE value of the shortest paths connecting nodes  $x$  and  $y$ . If nodes  $x$  and  $y$  are not connected, their minimum path ACE metric is defined as  $\infty$ . The ACE spectrum [13] of depth  $d_{max}$  of a TG  $G$  is defined as a  $d_{max}$ -tuple  $\boldsymbol{\eta}(G) = (\eta_2, \eta_4, \dots, \eta_{2d_{max}})$ , where  $\eta_{2i}$ ,  $1 \leq i \leq d_{max}$  is the minimum ACE value of all  $2i$ -cycles in  $G$ . If  $G$  does not contain any  $2i$ -cycles,  $\eta_{2i}$  is set as  $\infty$ . Furthermore, the comparison rule between two ACE spectra of depth  $d_{max}$  in [14] is defined as  $\boldsymbol{\eta}^{(1)} > \boldsymbol{\eta}^{(2)} \iff \exists j \leq d_{max} \left| \left( \eta_{2j}^{(1)} > \eta_{2j}^{(2)} \text{ and } \eta_{2i}^{(1)} = \eta_{2i}^{(2)}, 1 \leq i < j \right) \right.$ . In such case, the TG with a larger ACE spectrum is generally considered to be better [13]–[16].

### D. Other Notations and Definitions

In the rest, the following notations are used. Denote:

- $f_{x,y}^{(G)}$ ,  $\forall x, y \in V$  as the *metric* between nodes  $x$  and  $y$  under  $G$ . In this paper, each time of using  $f_{x,y}^{(G)}$  always refers to one of the following two metrics:

$$f_{x,y}^{(G)} = d_{x,y}^{(G)}, \quad (1)$$

$$f_{x,y}^{(G)} = (d_{x,y}^{(G)}, a_{x,y}^{(G)}), \quad (2)$$

where  $d_{x,y}^{(G)}$  indicates the distance and  $a_{x,y}^{(G)}$  indicates the minimum path ACE metric between nodes  $x$  and  $y$  under  $G$  respectively;

- $F_{X,y}^{(G)} = \{f_{x,y}^{(G)} | \forall x \in X\}, \forall X \subseteq V, \forall y \in V$ ;
- $F_{X,Y}^{(G)} = \{f_{x,y}^{(G)} | \forall x \in X, \forall y \in Y\}, \forall X, Y \subseteq V$ ;
- $\mathbf{0} = 0, \mathbf{1} = 1, \infty = \infty, \mathbf{0}_{v_j} = 0, \mathbf{1}_{v_j} = 1$  if the metric is defined as (1), otherwise  $\mathbf{0} = (0, 0), \mathbf{1} = (1, 0), \infty = (\infty, \infty), \mathbf{0}_{v_j} = (0, d_{v_j} - 2), \mathbf{1}_{v_j} = (1, d_{v_j} - 2)$  if the metric is defined as (2), where  $v_j, 0 \leq j < n$  is the  $j$ -th VN of the TG;

- $g^{(G)}$  as the *girth* of  $G$ , indicating the metric of the minimum cycle in  $G$ ; (When the metric of any specific cycle is referred to, if there no corresponding cycle exists, the metric is regarded as  $\infty$ .)
- $g_x^{(G)}$ ,  $\forall x \in V$  as the *local girth* of node  $x$  under  $G$ , indicating the metric of the minimum cycle in  $G$  which passes through  $x$ ;
- $g_{(x,y)}^{(G)}$ ,  $\forall x \in V_c, \forall y \in V_v$  as the local girth of edge  $(x, y)$  under  $G$ , indicating the metric of the minimum cycle in  $G$  which passes through  $(x, y)$ ;
- $|X|$  as the cardinality of an arbitrary set  $X$ ;
- $N$  as the one-dimension circulant-size of  $\mathbf{H}$ ; (I.e., each circulant of  $\mathbf{H}$  has size  $N \times N$ .)
- $\{(c_i, v_j)_N\} = \{(c_{\pi(i,N,t)}, v_{\pi(j,N,t)}) | t = 0, 1, \dots, N-1\}$  for  $0 \leq i < m$  and  $0 \leq j < n$ , where  $\pi(x, N, t) = \lfloor x/N \rfloor \cdot N + \text{mod}(x+t, N)$ , while  $\lfloor x/N \rfloor$  is the floor of  $x/N$  and  $\text{mod}(x+t, N)$  is the remainder of  $(x+t)$  modulo  $N$ .

In addition, in this paper, the comparison rule between two pairs of (2) is defined as  $f_{x_0,y_0}^{(G)} > f_{x_1,y_1}^{(G)} \iff d_{x_0,y_0}^{(G)} > d_{x_1,y_1}^{(G)}$  or  $(d_{x_0,y_0}^{(G)} = d_{x_1,y_1}^{(G)} \text{ and } a_{x_0,y_0}^{(G)} > a_{x_1,y_1}^{(G)})$ . The addition/subtraction between two pairs of (2) is defined as  $f_{x_0,y_0}^{(G)} \pm f_{x_1,y_1}^{(G)} = (d_{x_0,y_0}^{(G)} \pm d_{x_1,y_1}^{(G)}, a_{x_0,y_0}^{(G)} \pm a_{x_1,y_1}^{(G)})$ . Furthermore, to measure the VN-local-girth distribution (VNLGD) of a TG under the metric (1), a polynomial  $\phi(x) = \sum_{i>0} p_i x^i$  is defined, where  $p_i$  has indicated the percentage of the VNs whose local girths are  $i$  among all the VNs. If there are no VNs whose local girths are  $i$ ,  $p_i$  is considered as 0 and  $p_i x^i$  will be omitted from  $\phi(x)$ . Meanwhile, the comparison rule between two VNLGDs is defined as  $\phi^{(1)}(x) < \phi^{(2)}(x) \iff \exists j > 0 \left( p_j^{(1)} < p_j^{(2)} \text{ and } p_i^{(1)} = p_i^{(2)}, i < j \right)$ . In such case, the TG with a smaller VNLGD is considered to be better with regard to that it generally contains less small (may be the smallest) cycles. Finally, an operation  $\uplus$ , which works between a TG  $G$  and an edge  $e$  or an edge-set  $\tilde{E}$ , is defined as  $G \uplus e = (V, E \cup \{e\})$  or  $G \uplus \tilde{E} = (V, E \cup \tilde{E})$ .

### E. PEG Algorithm and ACE Constrained PEG Algorithm

The PEG algorithm [10] constructs the TG by establishing edges between VNs and CNs in stages under the metric (1), where in each stage only one edge is established. The edges are established in the order from small to large based on the indices of the VNs they are incident to. To be specific,  $v_0$  takes the first  $d_{v_0}$  consecutive stages to establish its edges, and then  $v_1$  takes the next  $d_{v_1}$  consecutive stages to establish its edges, and the construction process continues until  $v_{n-1}$  takes the last  $d_{v_{n-1}}$  stages to establish its edges. In such case, the VN, which is of interest to any stage, will be known before starting the construction process. Then, the VN, which is

of interest to the current stage, is conveniently denoted as  $v_c$ . Furthermore, by saying the  $k$ -th stage of  $v_c$ , where  $1 \leq k \leq d_{v_c}$  always holds and sometimes will be omitted in the rest of this paper, we refer to the stage in which the  $k$ -th edge of  $v_c$  is established.

In each stage of  $v_c$ , a CN, say  $c_i$ , is selected based on the *selection strategy* at the beginning of this stage, and then  $(c_i, v_c)$  is established at the end of this stage. The selection strategy is a list of selection criteria of decreasing priority [14]. If multiple CNs survive any selection criterion, the next selection criterion is applied on the surviving CNs only. Otherwise, the single survivor is selected and the selection procedure is terminated. Let  $G$  be the realtime TG setting, then the selection strategy of the PEG algorithm [10] is summarized in the following Strategy 1, while the pseudo-code of the PEG algorithm [10] is presented in Algorithm 1.

*Strategy 1 (Selection strategy of the PEG algorithm [10]):*

- 1) Select the CN  $c_i$  that  $f_{c_i, v_c}^{(G)} = \max_{0 \leq j < m} f_{c_j, v_c}^{(G)}$ ;
- 2) Select the survivor whose degree is minimal;
- 3) Select the survivor randomly.

---

**Algorithm 1** The PEG Algorithm [10]

---

**Input:**  $m, n, D$ .

**Output:**  $G$ .

- 1:  $G = (V, \emptyset)$ .
  - 2: **for**  $j = 0$ ;  $j < n$ ;  $j++$  **do**
  - 3:    $f_{c_i, v_j}^{(G)} = \infty$  for  $0 \leq i < m$ .
  - 4:   **for**  $k = 1$ ;  $k \leq d_{v_j}$ ;  $k++$  **do**
  - 5:      $c_i$  = the CN selected based on Strategy 1.
  - 6:      $G = G \uplus (c_i, v_j)$ .
  - 7:     Calculate  $F_{V_c, v_j}^{(G)}$ .
  - 8:   **end for**
  - 9: **end for**
  - 10: **return**  $G$ .
- 

*Remark 1:* Assuming that  $c_i$  is selected based on Strategy 1, according to Corollary 3 (refer to Section III-B), it holds that  $g_{v_c}^{(G \uplus (c_i, v_c))} = g_{(c_i, v_c)}^{(G \uplus (c_i, v_c))} = f_{c_i, v_c}^{(G)} + 1$ , implying that the establishment of  $(c_i, v_c)$  makes  $v_c$  achieve its maximum realtime local girth.

*Remark 2:* In Algorithm 1, line 7 is the most time-consuming part. Hu *et al.* [10] employed a breadth-first search (BFS) [17] like method to implement this calculation. In the worst case, the computational complexity of implementing the BFS once in  $G$  is  $O(m + |E|)$ . As the calculation is totally implemented  $|E|$  times, the total computational complexity of the original PEG algorithm [10] thus is  $O(m|E| + |E|^2)$ .

The key idea of the ACE constrained PEG algorithm<sup>1</sup> [14] is to construct the LDPC codes under the framework of the PEG algorithm [10], (Refer to Strategy 1 and Algorithm 1.) while replacing the metric (1) with the metric (2). However, the modification on the metric makes the ACE constrained PEG algorithm [14] a better algorithm, compared to the PEG algorithm [10], the ACE algorithm [11], *etc.*, in terms of designing the LDPC codes with better ACE spectra and better error performances. In addition, the total computational complexity of the ACE constrained PEG algorithm [14] remains the same as that of the PEG algorithm [10], i.e.,  $O(m|E| + |E|^2)$ .

#### F. Circulant, CPM, and QC-LDPC Code

A circulant is a square matrix where each row-vector is cyclically shifted one element to the right relative to the preceding row-vector. (The first row-vector's preceding row-vector is the last row-vector.) Consequently, a zero square matrix is also regarded as a circulant, i.e., the zero circulant. If there is only one entry of 1 in each row and each column of a circulant and 0s elsewhere, the circulant is considered as a CPM.

If the check matrix of an LDPC code is an array of sparse circulants with the same size, then it is a QC-LDPC code. The check matrix of a QC-LDPC code typically looks like follows:

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{0,0} & \mathbf{H}_{0,1} & \cdots & \mathbf{H}_{0,K-1} \\ \mathbf{H}_{1,0} & \mathbf{H}_{1,1} & \cdots & \mathbf{H}_{1,K-1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{H}_{J-1,0} & \mathbf{H}_{J-1,1} & \cdots & \mathbf{H}_{J-1,K-1} \end{bmatrix}, \quad (3)$$

where  $\mathbf{H}_{i,j}$ ,  $0 \leq i < J$ ,  $0 \leq j < K$  are circulants of the same size. Furthermore, in the TG  $G$  of a QC-LDPC code, for  $0 \leq i < m$ ,  $0 \leq j < n$ , it obviously holds that

$$(c_i, v_j) \in E \iff \{(c_i, v_j)_N\} \subseteq E. \quad (4)$$

<sup>1</sup>The ACE constrained PEG algorithm introduced in this paper is a little different from the original one in [14]. See [14] for more details.



### G. QC-PEG Algorithm

The QC-PEGA [33] constructs the QC-LDPC codes with any predefined valid design parameters. Li *et al.* [33] implemented the QC-PEGA similar to the PEG algorithm [10], where both the metric and the selection strategy remain the same. However, because of (4), the QC-PEGA [33] cyclically establishes  $N$  edges in a single circulant at a time in each stage, instead of only one edge in each stage as that in the PEG algorithm [10]. To be specific, the pseudo-code of the QC-PEGA [33] is presented in Algorithm 2.

---

**Algorithm 2** The QC-PEG Algorithm [33]

---

**Input:**  $m, n, N, D$ .

**Output:**  $G$ .

```

1:  $G = (V, \emptyset)$ .
2: for  $j = 0; j < n; j += N$  do
3:    $f_{c_i, v_j}^{(G)} = \infty$  for  $0 \leq i < m$ .
4:   for  $k = 1; k \leq d_{v_j}; k++$  do
5:      $c_i$  = the CN selected based on Strategy 1.
6:      $G = G \uplus \{(c_i, v_j)_N\}$ .
7:     Calculate  $F_{V_c, v_j}^{(G)}$ .
8:   end for
9: end for
10: return  $G$ .
```

---

*Remark 3:* Algorithm 2 generates the check matrix of a QC-LDPC code with the form (3), where  $J = m/N$ ,  $K = n/N$ , and each circulant has size  $N \times N$ . Therefore, input parameters (design parameters) are considered valid if and only if: 1)  $m$  and  $n$  are multiples of  $N$ . 2)  $0 < d_{v_j} = d_{v_{j+1}} = \dots = d_{v_{j+N-1}} \leq m$  for  $j = 0, N, \dots, n - N$ .

*Remark 4:* In line 6 of Algorithm 2,  $N$  edges are established in a single circulant at a time. But the QC-PEGA [33] cannot real timely detect the cycles which contain two or more newly established edges. As a result, the QC-PEGA [33] sometimes results in 4-cycles just in a single circulant of the check matrix when there contain multiple edges. A typical example is given in Fig. 1(a), where  $c_2$  has the chance to survive Strategy 1 as well as to be selected for establishing the second edge of  $v_0$ , and 4-cycles will form in the single circulant if edges in  $\{(c_2, v_0)_4\}$  are established. To avoid generating 4-cycles in a single circulant of the check matrix, the CP-PEGA

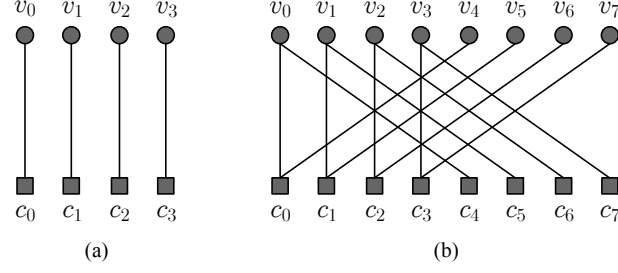


Fig. 1. Undetectable shortest cycles resulted by the QC-PEGA [33] and the CP-PEGA [34]. (a) Undetectable 4-cycles resulted by the QC-PEGA [33]. (b) Undetectable 8-cycles resulted by the CP-PEGA [34].

[34] requires each nonzero circulant of the check matrix to be a CPM during the construction process of the QC-PEGA [33]. However, this modification cannot real timely detect the cycles which contain two or more newly established edges either. Fig. 1(b) shows an example of such case, where  $c_6$  has the chance to survive Strategy 1 as well as to be selected for establishing the second edge of  $v_4$ , and 8-cycles will form if edges in  $\{(c_6, v_4)_4\}$  are established. In addition, it can be easily proved that 8 is the shortest length of the undetectable cycles, which contain two or more newly established edges, in the QC-LDPC code designed by the CP-PEGA [34]. Furthermore, because of the CPM-requirement, the CP-PEGA [34] additionally requires the maximum CN-degree not to exceed  $m/N$  (the number of circulants in a column), or it will fail the construction.

*Remark 5:* The construction process of Algorithm 2 is accelerated by a factor of  $1/N$  compared to that of Algorithm 1. Thus, the total computational complexity of the QC-PEGA [33] is  $O((m|E| + |E|^2)/N)$ .

### III. MULTI-EDGE METRIC-CONSTRAINED PEG ALGORITHM

#### A. MM-PEGA

Since the PEG algorithm [10] and the ACE constrained PEG algorithm [14] only differ at their metrics, this paper unifies them to one integrated algorithm, called the M-PEGA, where the PEG algorithm [10] is referred to when the metric is defined as (1), and the ACE constrained PEG algorithm [14] is referred to when the metric is defined as (2). As in each stage of the M-PEGA, a CN is selected to primarily maximize the local girth of  $v_c$  whenever a new edge is added to  $v_c$ . In such case, the final local girth of  $v_c$ , which is measured at the end of the  $d_{v_c}$ -th stage of  $v_c$ , may not be optimal. This situation is demonstrated in the following example.

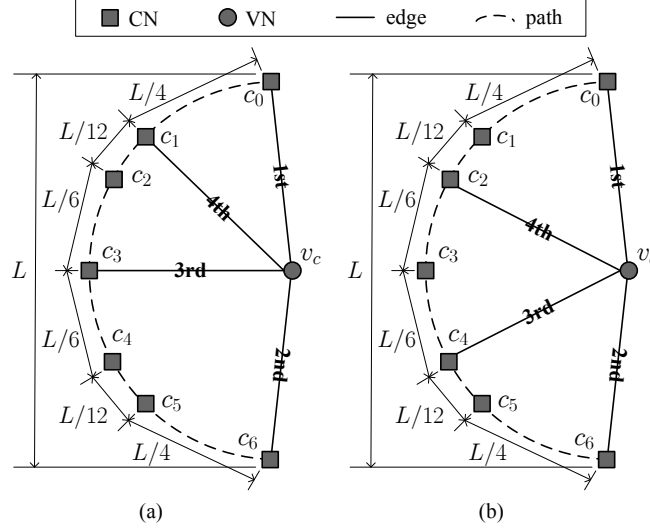


Fig. 2. The final local girth of  $v_c$  under the metric (1) achieved by the M-PEGA vs that achieved by the two-edge M-PEGA. (a) Edges are established by the M-PEGA. (b) Edges are established by the two-edge M-PEGA.

*Example 1:* Fig. 2 presents two simple TGs. At the beginning, each of the TGs consists of the VN  $v_c$  and the path connecting CNs  $c_0$  and  $c_6$ . Then, four edges are established between  $v_c$  and some CNs in the path by some specific design algorithm. In addition, only CNs  $c_0$ – $c_6$  are displayed in the path of each TG, while the other CNs are omitted since they have no chances to be selected for establishing the edges of  $v_c$  under the considered design algorithms. Assume that the metric is defined as (1), and then denote the lengths (distances) between different pairs of CNs in each path as expressions of  $L$ . As Fig. 2 aims to compare the relative sizes between the final local girths of  $v_c$  under different design algorithms, only assume that  $L$  is a valid positive integer while its value is not in the interest. Fig. 2(a) shows how the M-PEGA establishes the four edges of  $v_c$ , where the edges are labeled 1st, 2nd, 3rd, and 4th based on the order of being established. According to Strategy 1, in Fig. 2(a), the local girth of  $v_c$  has been maximized whenever a new edge is added to  $v_c$ . In contrast, another case is shown in Fig. 2(b). (The construction process will be shown in detail in Example 2.) The establishment of the third edge in Fig. 2(b) is considered to be worse than that in Fig. 2(a) in terms of making the realtime local girth of  $v_c$  maximum, where  $g_{v_c}^{(G_b)} = L/3 + 2$  in Fig. 2(b) vs  $g_{v_c}^{(G_a)} = L/2 + 2$  in Fig. 2(a) at the end of the third stage of  $v_c$ . (Here in this example,  $G_a$  and  $G_b$  temporarily represent the realtime TG settings in Fig. 2(a) and in Fig. 2(b) respectively.) Nonetheless, if it is known that where the fourth edge will be established at the beginning of the third stage of  $v_c$ , the establishment of the third edge in Fig. 2(b) is considered to be better than that in Fig. 2(a) in terms of making

the final local girth of  $v_c$  maximum, where  $g_{v_c}^{(G_b)} = L/3 + 2$  in Fig. 2(b) vs  $g_{v_c}^{(G_a)} = L/4 + 2$  in Fig. 2(a) at the end of the fourth stage of  $v_c$ .

Inspired by Example 1, this paper proposes an improvement for the M-PEGA, called the MM-PEGA, to further optimize the cycle-structure of the non-QC LDPC code. On the whole, the MM-PEGA is implemented under the framework of Algorithm 1 but adopts a different selection strategy. To illustrate the MM-PEGA as well as its selection strategy, the multi-edge local girths are defined firstly,

Assume that  $G = (V_c \cup V_v, E)$  is the current TG setting, and let  $r$  be a positive integer. Then, under  $G$  and  $v_c$ , a length- $r$  *CN-sequence (CNS)*  $\mathbf{S} = (s_i)_{1 \leq i \leq r}$ , where  $s_i \in V_c$  for  $1 \leq i \leq r$ , is called *valid* if  $s_i \neq s_j$  for  $1 \leq i < j \leq r$  and  $(s_i, v_c) \notin E$  for  $1 \leq i \leq r$ . In addition, the corresponding *TG-sequence (TGS)* of  $\mathbf{S}$  is defined as  $(G_i)_{1 \leq i \leq r+1}$ , where  $G_1 = G$  and  $G_{i+1} = G_i \uplus (s_i, v_c)$  for  $1 \leq i \leq r$ .

*Definition 1:* Under  $G$ ,  $v_c$ , and  $r$ , let  $\mathbf{S}_{v_c}^{(r,G)}$  be a length- $r$  valid CNS with its corresponding TGS  $(G_i^{(\mathbf{S})})_{1 \leq i \leq r+1}$ . Then, the  $r$ -edge local girth of  $v_c$  is defined as

$$g_{v_c}^{(r,G)} = \max_{\mathbf{S}_{v_c}^{(r,G)}} g_{v_c}^{(G_{r+1}^{(\mathbf{S})})}, \quad (5)$$

and the  $r$ -edge local girth of  $c_i$ ,  $0 \leq i < m$  is defined as

$$g_{c_i, v_c}^{(r,G)} = \begin{cases} -\infty & \text{if } (c_i, v_c) \in E, \\ g_{v_c}^{(G \uplus (c_i, v_c))} & \text{else if } r = 1, \\ g_{v_c}^{(r-1, G \uplus (c_i, v_c))} & \text{otherwise.} \end{cases} \quad (6)$$

In (5),  $g_{v_c}^{(r,G)}$  indicates the maximum local girth of  $v_c$  measured after any  $r$  edges in  $E_{v_c} \setminus E$  will have been established. (All the  $r$  edges are incident to  $v_c$  while have not been established in  $G$ .) Naturally, for some length- $r$  valid CNS  $\hat{\mathbf{S}}_{v_c}^{(r,G)}$  with corresponding TGS  $(\hat{G}_i)_{1 \leq i \leq r+1}$ , if  $g_{v_c}^{(r,G)} = g_{v_c}^{(\hat{G}_{r+1})}$ , then  $\hat{\mathbf{S}}_{v_c}^{(r,G)}$  is called *optimal* for  $g_{v_c}^{(r,G)}$ . In (6), if  $(c_i, v_c) \notin E$ ,  $g_{c_i, v_c}^{(r,G)}$  indicates the maximum local girth of  $v_c$  measured after  $(c_i, v_c)$  as well as any  $r-1$  edges in  $E_{v_c} \setminus E \setminus \{(c_i, v_c)\}$  will have been established. It obviously holds that

$$g_{v_c}^{(r,G)} = \max_{0 \leq i < m} g_{c_i, v_c}^{(r,G)} \quad \text{s.t.} \quad r > 0. \quad (7)$$

*Definition 2:* The *edge-trials* is a maximum number of edges tried by a specific MM-PEGA in each stage to construct a local optimal edge of the TG. The MM-PEGA with the edge-trials, say  $r$ , is called the  $r$ -edge M-PEGA.

TABLE I  
TWO-EDGE LOCAL GIRTHS OF CNS IN DIFFERENT STAGES OF THE CONSTRUCTION PROCESS IN EXAMPLE 2

CN	1st	2nd	3rd	4th
$c_0$	$L + 2$	$-\infty$	$-\infty$	$-\infty$
$c_1$	$3L/4 + 2$	$L/4 + 2$	$L/4 + 2$	$L/4 + 2$
$c_2$	$2L/3 + 2$	$L/3 + 2$	$L/3 + 2$	$L/3 + 2$
$c_3$	$L/2 + 2$	$L/2 + 2$	$L/4 + 2$	$L/6 + 2$
$c_4$	$2L/3 + 2$	$L/3 + 2$	$L/3 + 2$	$-\infty$
$c_5$	$3L/4 + 2$	$3L/8 + 2$	$L/4 + 2$	$L/12 + 2$
$c_6$	$L + 2$	$L/2 + 2$	$-\infty$	$-\infty$

The edge-trials of a specific MM-PEGA is closely related to its selection strategy. More precisely, under the current TG setting  $G$ , the following Strategy 2 is adopted by the  $r$ -edge M-PEGA to at the beginning of the  $k$ -th stage of  $v_c$ , where  $k$  is defined in line 4 of Algorithm 1, and  $r_k$  in Strategy 2 is defined as

$$r_k = \min \{r, d_{v_c} - k + 1\}. \quad (8)$$

*Strategy 2 (Selection strategy of the  $r$ -edge M-PEGA):*

- 1) Select the CN  $c_i$  that  $g_{c_i, v_c}^{(r_k, G)} = g_{v_c}^{(r_k, G)}$ ;
- 2) Select the survivor  $c_i$  that  $g_{(c_i, v_c)}^{(G \uplus (c_i, v_c))} = \max \left\{ g_{(x, v_c)}^{(G \uplus (x, v_c))} \middle| x \in V_c, g_{x, v_c}^{(r_k, G)} = g_{v_c}^{(r_k, G)} \right\}$ ;
- 3) Select the survivor whose degree is minimal;
- 4) Select the survivor randomly.

Since the multi-edge local girth of  $v_c$  should be considered under the premise of that  $v_c$  only has  $d_{v_c}$  edges to be established, (8) is thus reasonable. In addition, in Strategy 2, the first selection criterion selects the CN whose  $r_k$ -edge local girths is optimal, aiming to make  $v_c$  achieve its maximum local girth after  $r_k$  new edges are added to  $v_c$ . Then, from the CNs which have survived the first selection criterion, the second selection criterion selects a survivor to establish the edge with the largest local girth, aiming to avoid generating cycles with smaller sizes.

*Example 2:* This example illustrates how the four edges of  $v_c$  in Fig. 2(b) are established by the two-edge M-PEGA under the metric (1), where the edges are also labeled 1st, 2nd, 3rd, and 4th based on the order of being established. The two-edge local girths of CNs  $c_0$ – $c_6$  measured at the beginning of different stages of  $v_c$  are presented in Table I. For instance, the column labelled by 1st displays the two-edge local girths of CNs  $c_0$ – $c_6$  measured at the beginning of the first

stage of  $v_c$ , and the column labelled by 2nd displays the two-edge local girths of CNs  $c_0$ – $c_6$  measured at the beginning of the second stage of  $v_c$ , and so on. However, because of (8) and Strategy 2, the last column labelled by 4th displays the one-edge instead of the two-edge local girths of CNs  $c_0$ – $c_6$  measured at the beginning of the fourth stage of  $v_c$ . To be more specific, the following four stages show how the four edges of  $v_c$  are established in Fig. 2(b) in detail:

- 1) At the beginning of the first stage of  $v_c$ , only  $c_0$  and  $c_6$  survive the first selection criterion of Strategy 2. Furthermore,  $c_0$  and  $c_6$  perform the same during the following comparisons until  $c_0$  is selected at last by randomness. Then,  $(c_0, v_c)$  is established as the first edge of  $v_c$ .
- 2) At the beginning of the second stage of  $v_c$ , only  $c_3$  and  $c_6$  survive the first selection criterion of Strategy 2. However,  $c_6$  is selected in this stage since it defeats  $c_3$  on the second selection criterion of Strategy 2. Then,  $(c_6, v_c)$  is established as the second edge of  $v_c$ .
- 3) At the beginning of the third stage of  $v_c$ , only  $c_2$  and  $c_4$  survive the first selection criterion of Strategy 2. Furthermore,  $c_2$  and  $c_4$  perform the same during the following comparisons until  $c_4$  is selected at last by randomness. Then,  $(c_4, v_c)$  is established as the third edge of  $v_c$ .
- 4) Finally, at the beginning of the fourth stage of  $v_c$ , only  $c_2$  survives the first selection criterion of Strategy 2. Then,  $(c_2, v_c)$  is established as the fourth edge of  $v_c$ .

Fig. 2 presents a specific case to show that  $v_c$  may achieve a larger final local girth under the TG designed by the two-edge M-PEGA, compared to the final local girth achieved under the TG designed by the M-PEGA. In addition, the design under the two-edge M-PEGA in Fig. 2(b) is optimal (with regard to making the final local girth of  $v_c$  maximum). Furthermore, any MM-PEGA with an edge-trials larger than one can achieve the optimal design, while the one-edge M-PEGA achieves the design as same as that under the M-PEGA in Fig. 2(a). (It can be seen in Corollary 3 that the one-edge M-PEGA is equivalent to the M-PEGA.) However, the MM-PEGA with a larger edge-trials may not always achieve a better design. For example, change  $d_{v_c}$  to 5 in Fig. 2, then the final local girth of  $v_c$  under the two-edge M-PEGA will be  $2L/9 + 2$ , while any MM-PEGA with an edge-trials except two makes  $v_c$  achieve its optimal final local girth  $L/4 + 2$ . In general, under the same TG setting at the beginning of the first stage of  $v_c$ , since the  $d_{v_c}$ -edge M-PEGA always makes  $v_c$  achieve its maximum final local girth, it's expected that increasing the edge-trials of the MM-PEGA will have a positive effect on the final local girth of  $v_c$  as well as the cycle-structure of the target TG.

### B. Calculate the Multi-Edge Local Girths

Since the  $r$ -edge M-PEGA changes the Strategy 1 used in line 5 of Algorithm 1 to the Strategy 2, the multi-edge local girth of  $v_c$  as well as that of all the CNs need to be calculated between line 4 and line 5 of Algorithm 1, right before each time of applying Strategy 2. Before proposing the specific algorithm for calculating the multi-edge local girths, we would like to propose some propositions and corollaries to discuss the properties of the multi-edge local girths firstly.

Assume that the TG has been constructed by the  $r$ -edge M-PEGA. For  $1 \leq k \leq d_{v_c}$ , denote  $\hat{c}_k$  as the CN selected in the  $k$ -th stage of  $v_c$ , and denote  $(\hat{G}_i)_{1 \leq i \leq d_{v_c}+1}$  as the corresponding TGS of  $(\hat{c}_i)_{1 \leq i \leq d_{v_c}}$ , where  $\hat{G}_1$  denotes the TG that at the beginning of the first stage of  $v_c$ . Then, under  $\hat{G}_k$ , denote  $\mathbf{S}^{(k)} = (s_i^{(k)})_{1 \leq i \leq r_k}$  as an arbitrary length- $r_k$  valid CNS with its corresponding TGS  $(G_i^{(k)})_{1 \leq i \leq r_k+1}$ , where  $G_1^{(k)} = \hat{G}_k$  obviously holds. Finally, define  $\alpha_k = \min_{1 \leq i < k} g_{(\hat{c}_i, v_c)}^{(\hat{G}_{i+1})}$  and  $\beta_k = \min_{1 \leq i \leq r_k} g_{(s_i^{(k)}, v_c)}^{(G_{i+1}^{(k)})}$ .

*Proposition 1:* For  $1 \leq k \leq d_{v_c}$ , it holds that  $g_{v_c}^{(r_k, \hat{G}_k)} = g_{\hat{c}_k, v_c}^{(r_k, \hat{G}_k)} \geq g_{v_c}^{(G_{r_k+1}^{(k)})} = \min \{\alpha_k, \beta_k\} = \beta_k$ .

*Proof:* For  $1 \leq k \leq d_{v_c}$ , it obviously holds that  $g_{v_c}^{(r_k, \hat{G}_k)} = g_{\hat{c}_k, v_c}^{(r_k, \hat{G}_k)} \geq g_{v_c}^{(G_{r_k+1}^{(k)})} = \min \{\alpha_k, \beta_k\}$ . In addition, for  $k = 1$ ,  $\min \{\alpha_k, \beta_k\} = \beta_k$  obviously holds. Instead, for  $2 \leq k \leq d_{v_c}$ , by assuming that  $\alpha_k < \beta_k$ , a contradiction is found in the following proof.

According to the assumption that  $\alpha_k < \beta_k$ , there exists some index  $i, 1 \leq i < k$  satisfying  $\alpha_k = g_{(\hat{c}_i, v_c)}^{(\hat{G}_{i+1})} \geq g_{\hat{c}_i, v_c}^{(r_i, \hat{G}_i)} \geq g_{\hat{c}_k, v_c}^{(r_i, \hat{G}_i)} \geq g_{\hat{c}_k, v_c}^{(r_{i+1}, \hat{G}_{i+1})} \geq \dots \geq g_{\hat{c}_k, v_c}^{(r_k, \hat{G}_k)} \geq g_{v_c}^{(G_{r_k+1}^{(k)})} = \min \{\alpha_k, \beta_k\} = \alpha_k$ , where the 1st and the 3rd “ $\geq$ ” hold because of (6) and (8), and the 2nd “ $\geq$ ” holds because of Strategy 2. Consequently, it holds that  $g_{(\hat{c}_i, v_c)}^{(\hat{G}_{i+1})} = g_{\hat{c}_i, v_c}^{(r_i, \hat{G}_i)} = g_{\hat{c}_k, v_c}^{(r_i, \hat{G}_i)} = g_{\hat{c}_k, v_c}^{(r_k, \hat{G}_k)} = g_{v_c}^{(G_{r_k+1}^{(k)})} = \alpha_k$ .

As a result, on the one hand, both  $\hat{c}_i$  and  $\hat{c}_k$  survive the first selection criterion of Strategy 2 at the beginning of the  $i$ -th stage of  $v_c$ . On the other hand, it holds that

$$g_{(\hat{c}_k, v_c)}^{(\hat{G}_i \uplus (\hat{c}_k, v_c))} \geq g_{(\hat{c}_k, v_c)}^{(\hat{G}_{i+1})} \geq \max_{1 \leq i \leq r_k} g_{(s_i^{(k)}, v_c)}^{(\hat{G}_k \uplus (s_i^{(k)}, v_c))} \geq \max_{1 \leq i \leq r_k} g_{(s_i^{(k)}, v_c)}^{(G_{i+1}^{(k)})} \geq \beta_k > \alpha_k = g_{(\hat{c}_i, v_c)}^{(\hat{G}_{i+1})}, \quad (9)$$

where the 2nd “ $\geq$ ” holds because each CN in  $\{\hat{c}_k, s_1^{(k)}, s_2^{(k)}, \dots, s_{r_k}^{(k)}\}$  survives the first selection criterion of Strategy 2 while  $\hat{c}_k$  continuously survives the second selection criterion of Strategy 2 at the beginning of the  $k$ -th stage of  $v_c$ .

Therefore, in the  $i$ -th stage of  $v_c$ ,  $\hat{c}_k$  defeats  $\hat{c}_i$  with regard to the second selection criterion of Strategy 2. Consequently,  $\hat{c}_i$  has no chance to be selected for establishing the  $i$ -th edge of  $v_c$ ,

which results in a contradiction. So, the assumption of  $\alpha_k < \beta_k$  must be false, indicating that  $\min\{\alpha_k, \beta_k\} = \beta_k$ . Till now, Proposition 1 has been proved.  $\blacksquare$

*Corollary 1:* For  $1 \leq k \leq d_{v_c}$ , if  $\mathbf{S}^{(k)}$  is optimal for  $g_{v_c}^{(r_k, \hat{G}_k)}$ , it holds that  $g_{v_c}^{(r_k, \hat{G}_k)} = g_{\hat{c}_k, v_c}^{(r_k, \hat{G}_k)} = g_{v_c}^{(G_{r_k+1}^{(k)})} = \min\{\alpha_k, \beta_k\} = \beta_k$ .

Corollary 1 can be easily proved based on Proposition 1. In addition, Proposition 1 implies that the final local girth of  $v_c$  is always sufficiently upper-bounded by the realtime local girths of the edges which are established in  $v_c$ 's later stages. Meanwhile, Corollary 1 implies that currently selecting a CN, whose multi-edge local girth is optimal, is expected to have a positive effect on the final local girth of  $v_c$ , and this is exactly what the first selection criterion of Strategy 2 does for. To conveniently calculate the multi-edge local girths, the following corollaries are proposed.

*Corollary 2:* Assume that the current TG setting  $G$  has been constructed by the  $r$ -edge M-PEGA, and let  $\mathbf{S}_{v_c}^{(r, G)} = (s_i)_{1 \leq i \leq r}$  be a length- $r$  valid CNS with its corresponding TGS  $(G_i^{(\mathbf{S})})_{1 \leq i \leq r+1}$ , then it holds that

$$g_{v_c}^{(r, G)} = \max_{\mathbf{S}_{v_c}^{(r, G)}} \left( \min_{1 \leq i \leq r} g_{(s_i, v_c)}^{(G_{i+1}^{(\mathbf{S})})} \right), \quad (10)$$

$$g_{v_c}^{(r, G)} = \max_{\mathbf{S}_{v_c}^{(r, G)}} \left( \min_{1 \leq i \leq r} \left( f_{s_i, v_c}^{(G_i^{(\mathbf{S})})} + 1 \right) \right). \quad (11)$$

*Corollary 3:* Assuming that the current TG setting  $G$  has been constructed by the one-edge M-PEGA, it holds that  $g_{v_c}^{(1, G)} = \max_{0 \leq i < m, (c_i, v_c) \notin E} g_{(c_i, v_c)}^{(G \uplus (c_i, v_c))} = \max_{0 \leq i < m} (f_{c_i, v_c}^{(G)} + 1)$ .

Corollary 2 can be easily proved based on (5) and Corollary 1, while Corollary 3 is directly deduced from Corollary 2. Consequently, the one-edge M-PEGA is equivalent to the M-PEGA when the metrics are defined as the same. Furthermore, this paper employs a DFS like algorithm, presented in Algorithm 3, to calculate the  $r$ -edge local girths in the way of (11).

*Remark 6:* Before starting Algorithm 3,  $g_{v_c}^{(r, G)}$  and  $g_{c_i, v_c}^{(r, G)}$ ,  $0 \leq i < m$  need to be set as  $-\infty$ .

*Remark 7:* In Algorithm 3,  $t, G_t, g_t, u_t$  are local variables of the  $t$ -th (starting from 1) layer, where  $t = 1, G_1 = G, g_1 = \infty, u_1 = m$  in the first layer. Besides, other variables are global except the  $i$  in line 2.

*Remark 8:*  $u_t$  works as an index-upper-bound of the enumerated CNs of the  $t$ -th layer, making the indices of CNs in each enumerated CNS strictly decreasing. In such case, redundant enumerations of CNSs have been avoided effectively.

*Remark 9:* Line 7 of Algorithm 3 cuts down useless search for  $g_{v_c}^{(r, G)}$ , making the DFS process much more time-efficient according to the simulation results.



**Algorithm 3** DFS Calculation of the  $r$ -Edge Local Girths**Input:**  $t, G_t = (V_c \cup V_v, E_t), g_t, u_t$ .

---

```

1:  $\lambda_t = -\infty$ .
2: for  $i = 0; i < u_t; i++$  do
3:   if  $(c_i, v_c) \notin E_t$  then
4:      $G_{t+1} = G_t \uplus (c_i, v_c)$ .
5:      $g_{(c_i, v_c)}^{(G_{t+1})} = f_{c_i, v_c}^{(G_t)} + 1$ .
6:      $g_{t+1} = \min \left\{ g_t, g_{(c_i, v_c)}^{(G_{t+1})} \right\}$ .
7:     if  $g_{t+1} \geq g_{v_c}^{(r, G)}$  then
8:       if  $t == r$  then
9:          $g_{v_c}^{(r, G)} = g_{c_i, v_c}^{(r, G)} = \lambda_t = g_{t+1}$ .
10:      else
11:         $u_{t+1} = i$ .
12:        Calculate  $F_{V_c, v_c}^{(G_{t+1})}$ .
13:        Enter the  $(t+1)$ -th layer with input parameters  $t+1, G_{t+1}, g_{t+1}, u_{t+1}$  respectively,
        and return here after the calculation in the  $(t+1)$ -th layer is finished.
14:         $\lambda_t = \max \left\{ \lambda_t, \lambda_{t+1} \right\}$ .
15:         $g_{c_i, v_c}^{(r, G)} = \max \left\{ g_{c_i, v_c}^{(r, G)}, \lambda_{t+1} \right\}$ .
16:      end if
17:    end if
18:  end if
19: end for

```

---

*Remark 10:* The metric-calculations between  $V_c$  and  $v_c$  under the realtime TG settings (i.e., those in line 7 of Algorithm 1 and in line 12 of Algorithm 3) are the most time-consuming parts. On the one hand, before each time of entering the  $t$ -th,  $1 \leq t \leq r$  layer, the metric-calculation in the  $(t-1)$ -th layer<sup>2</sup> will be implemented once. On the other hand, the DFS process of Algorithm 3 will reach the  $t$ -th,  $1 \leq t \leq r$  layer at most  $m^{t-1}/(t-1)! \approx m^{t-1}$  (since generally  $r$  is small and  $r \ll m$ ) times. In such case, to apply Algorithm 3 in the  $r$ -edge M-PEGA once, the metric-calculation will be totally implemented at most  $\sum_{t=1}^r m^{t-1} = (m^r - 1)/(m - 1) \approx m^{r-1}$

<sup>2</sup>The metric-calculation in the 0-th layer indicates that in the line 7 of Algorithm 1.

(since generally  $m \gg 1$ ) times. In addition, the BFS like method used in the PEG algorithm [10] is employed to implement the metric-calculation. Since Algorithm 3 will be applied in the  $r$ -edge M-PEGA  $|E|$  times, thus the total computational complexity of the  $r$ -edge M-PEGA is  $O(m^r|E| + m^{r-1}|E|^2)$ .

#### IV. MULTI-EDGE METRIC-CONSTRAINED QC-PEG ALGORITHM

To design the QC-LDPC codes with any predefined valid design parameters (refer to Remark 3), as well as to detect and even to avoid generating the undetectable cycles in the QC-LDPC codes designed by the QC-PEGA [33] and the CP-PEGA [34], the MM-QC-PEGA is proposed in this section, which is implemented under the framework of Algorithm 2 but adopts a different selection strategy. Since the selection strategy of the MM-QC-PEGA is closely related to that of the MM-PEGA, as well as to simplify the illustration of the MM-QC-PEGA, the illustration of the MM-PEGA in last section has been employed here. In general, because of (4), each concept related to a single edge  $(c_i, v_j)$  in the MM-PEGA changes to a similar concept related to the edge-set  $\{(c_i, v_c)_N\}$  in the MM-QC-PEGA, while the other concepts remain the same.

To be specific, the definition for a length- $r$  valid CNS  $\mathbf{S} = (s_i)_{1 \leq i \leq r}$  remains the same, while its corresponding TGS  $(G_i)_{1 \leq i \leq r+1}$  changes, where  $G_1 = G$  too, but  $G_{i+1} = G_i \uplus \{(s_i, v_c)_N\}$  for  $1 \leq i \leq r$ . In such case, Definition 1 changes a little, where  $G \uplus (c_i, v_c)$  in (6) changes to  $G \uplus \{(c_i, v_c)_N\}$ . Then, (7) still holds for the MM-QC-PEGA. In addition, the edge-trials of the MM-QC-PEGA can be similarly defined as that in Definition 2. Finally, Strategy 2 with two modifications on its second selection criterion, where  $G \uplus (c_i, v_c)$  changes to  $G \uplus \{(c_i, v_c)_N\}$  and  $G \uplus (x, v_c)$  changes to  $G \uplus \{(x, v_c)_N\}$ , has been adopted by the MM-QC-PEGA.

To discuss the properties of the multi-edge local girths related to the QC-LDPC code, Proposition 1 has been employed too, which can be proved in almost the same way with only two small modifications in (9), where  $\hat{G}_i \uplus (\hat{c}_k, v_c)$  changes to  $\hat{G}_i \uplus \{(\hat{c}_k, v_c)_N\}$  and  $\hat{G}_k \uplus (s_i^{(k)}, v_c)$  changes to  $\hat{G}_k \uplus \{(s_i^{(k)}, v_c)_N\}$ . In addition, Corollary 1 as well as (10) in Corollary 2 still hold for the MM-QC-PEGA, but (11) may not hold any more because of the modification on the definition of the TGS. To explain this, the following illustration is given.

Assume that  $G = (V_c \cup V_v, E)$  is the TG of an arbitrary QC-LDPC code, and that  $(c_i, v_j)$  is an arbitrary edge satisfying  $c_i \in V_c$ ,  $v_j \in V_v$ , and  $\{(c_i, v_j)_N\} \cap E = \emptyset$ . In addition, let  $\tilde{G} = G \uplus \{(c_i, v_j)_N\}$  and  $\hat{G} = G \uplus (\{(c_i, v_j)_N\} \setminus \{(c_i, v_j)\})$ . Then, it holds that

$$g_{(c_i, v_j)}^{(\tilde{G})} = f_{c_i, v_j}^{(\hat{G})} + 1 \leq f_{c_i, v_j}^{(G)} + 1. \quad (12)$$

When  $N = 1$ ,  $f_{c_i, v_j}^{(\hat{G})} = f_{c_i, v_j}^{(G)}$  holds, implying that in such case the  $r$ -edge M-PEGA is equivalent to the  $r$ -edge M-QC-PEGA. However, when  $N > 1$ ,  $f_{c_i, v_j}^{(\hat{G})} < f_{c_i, v_j}^{(G)}$  may sometimes hold. For example, let  $G$  be the TG of Fig. 1(a), further assume that  $c_i = c_2$  and  $v_j = v_0$ , then  $f_{c_i, v_j}^{(\hat{G})} = 3 < f_{c_i, v_j}^{(G)} = \infty$ . Consequently, on the one hand, (12) explains the essential reasons that why the cycles, which contain two or more newly established edges, cannot be detected in the QC-LDPC codes designed by the QC-PEGA [33] and the CP-PEGA [34], but can be detected in the codes designed by the MM-QC-PEGA. On the other hand, Algorithm 3 needs to be modified when being applied in the MM-QC-PEGA to calculate the multi-edge local girths. Exactly, in Algorithm 3, line 4 changes to  $G_{t+1} = G_t \uplus \{(c_i, v_c)_N\}$ , and line 5 changes to

$$g_{(c_i, v_c)}^{(G_{t+1})} = f_{c_i, v_c}^{(\bar{G})} + 1, \quad (13)$$

where  $\bar{G} = G_t \uplus (\{(c_i, v_c)_N\} \setminus \{(c_i, v_c)\})$ .

To calculate  $f_{c_i, v_c}^{(\bar{G})}$  in (13), the BFS like method used in the original PEG algorithm [10] is employed. In such case, the computational complexity for calculating  $f_{c_i, v_c}^{(\bar{G})}$  once is  $O(m + |E|)$ , while the metric-calculation in line 7 of Algorithm 2 as well as that in line 12 of Algorithm 3 can be omitted. Since the calculation for  $f_{c_i, v_c}^{(\bar{G})}$  will be implemented approximately  $m^r$  times at each stage of the  $r$ -edge M-QC-PEGA according to Remark 10, thus the total computational complexity of the  $r$ -edge M-QC-PEGA is  $O((m^{r+1}|E| + m^r|E|^2)/N)$ .

## V. SIMULATION RESULTS

In this paper, a code is said to be regular if its VN-degrees are identical. Besides, the selection strategies of the aforementioned LDPC code design algorithms have tried to make the CN-degrees as identical as possible too. The bit error rate (BER) estimations of all simulated codes are performed using the Monte-Carlo simulation, assuming binary phase-shift keying (BPSK) transmission over the AWGN channel and the standard sum-product algorithm (SPA) iterative decoding with 100 decoding iterations at the receiver. In addition, at least 100 frame errors are captured at each simulated signal-to-noise ratio (SNR) point.

### A. Irregular LDPC Codes

*Example 3:* This example considers the design of the irregular (1008, 504) non-QC-LDPC codes using the MM-PEGA. The popular density evolution (DE) [36], [37] optimized VN-degree

TABLE II  
ACE SPECTRA OF DIFFERENT CODE SETS IN EXAMPLE 3

Code	Average	Maximum	Freq
$A_1$	$(\infty, \infty, 13.00, 6.08, 3.03)$	$(\infty, \infty, 13, 13, 4)$	0.0038
$A_2$	$(\infty, \infty, 13.00, 8.20, 3.03)$	$(\infty, \infty, 13, 13, 4)$	0.0130
$A_3$	$(\infty, \infty, 13.00, 12.53, 3.03)$	$(\infty, \infty, 13, 13, 4)$	0.0326
$A_4$	$(\infty, \infty, 13.00, 13.00, 3.03)$	$(\infty, \infty, 13, 13, 4)$	0.0295
$B_1$	$(\infty, \infty, 18.21, 8.85, 3.82)$	$(\infty, \infty, 26, 10, 4)$	0.0012
$B_2$	$(\infty, \infty, 18.53, 9.61, 3.88)$	$(\infty, \infty, 26, 13, 5)$	0.0006
$B_3$	$(\infty, \infty, 19.12, 12.69, 4.66)$	$(\infty, \infty, 26, 13, 5)$	0.0313
$B_4$	$(\infty, \infty, 20.44, 13.00, 5.08)$	$(\infty, \infty, 26, 13, 6)$	0.0229

TABLE III  
ACE SPECTRA OF THE SELECTED CODES IN EXAMPLE 3 AND THE FREQUENCIES ASSOCIATED WITH THE FORMER ACE SPECTRA

Code	Spectrum	Freq
$A_1$	$(\infty, \infty, 13, 13, 3)$	0.1229
$A_2$	$(\infty, \infty, 13, 13, 4)$	0.0130
$A_3$	$(\infty, \infty, 13, 13, 4)$	0.0326
$A_4$	$(\infty, \infty, 13, 13, 4)$	0.0295
$B_1$	$(\infty, \infty, 19, 10, 4)$	0.0259
$B_2$	$(\infty, \infty, 19, 13, 4)$	0.0128
$B_3$	$(\infty, \infty, 26, 13, 5)$	0.0313
$B_4$	$(\infty, \infty, 26, 13, 6)$	0.0229

distribution<sup>3</sup>  $\gamma(x) = 0.47532x^2 + 0.27953x^3 + 0.03486x^4 + 0.10889x^5 + 0.10138x^{15}$  is employed to design the codes. For convenience, the code (or code set) designed by the  $r$ -edge M-PEGA with the metric (1) is denoted as  $A_r$ , and that designed by the  $r$ -edge M-PEGA with the metric (2) is denoted as  $B_r$ , where  $r = 1, 2, 3, 4$  for both  $A_r$  and  $B_r$ .

More than 1000 codes of each code set are randomly constructed. The average ACE spectrum, the maximum ACE spectrum, and the frequency<sup>4</sup> associated with the maximum ACE spectrum of each code set are presented in Table II. In Table II, for the code sets designed with the metric (1)

<sup>3</sup>Assume that the VN-degree distribution of an LDPC code is  $\gamma(x) = \sum_i p_i x^i$ , then  $p_i$  has indicated the percentage of the degree- $i$  VNs among all the VNs.

<sup>4</sup>The frequency associated with an ACE spectrum indicates the frequency that the assumed ACE spectrum occurs among all the spectra of the codes from a same code set.

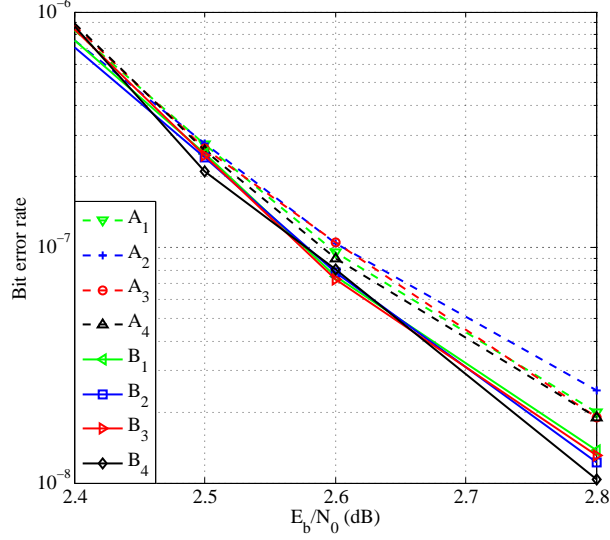


Fig. 3. BER performances of the irregular (1008, 504) LDPC codes designed in Example 3.

(i.e.,  $A_1$ – $A_4$ ), it can be seen that the average ACE spectrum increases as the edge-trials increases, but different code sets achieve the same maximum ACE spectrum. In general, the code set with a larger edge-trials can achieve the maximum ACE spectrum with a higher frequency. At the same time, for the code sets designed with the metric (2) (i.e.,  $B_1$ – $B_4$ ), it can be seen that both the average and the maximum ACE spectra increase as the edge-trials increases. In addition, it's consistent with the simulation results in [14] that the ACE spectra of the codes designed with the metric (2) greatly surpass that of the codes designed with the metric (1). Furthermore, compared to the ACE spectra of the LDPC codes constructed with the similar design parameters in the literature [13]–[16], the maximum ACE spectra of  $B_1$ – $B_4$  reported in Table II have been the best till now.

Two codes are randomly selected from each code set, keeping the ACE spectrum of each selected code maximum with an associated frequency larger than  $0.01^5$ . The ACE spectra of these selected codes as well as the frequencies associated with the former ACE spectra are presented in Table III. In addition, the BER estimations of these codes are performed. Then, for the two codes from a same code set, the one which has achieved a lower BER at  $E_b/N_0 = 2.8\text{dB}$  is remained. Furthermore, the error performances of the remained codes are presented in Fig.

<sup>5</sup>This threshold is chosen with two considerations: Firstly, each code set offers at least 10 candidate codes; Secondly, the ACE spectrum should be large enough so that the chance to get a code with the same ACE spectrum by reconstructing codes of the same code set will not be too small.

TABLE IV  
ACE SPECTRA OF DIFFERENT CODE SETS IN EXAMPLE 4

Code	Average	Maximum	Freq
$C_{1,0}$	$(\infty, -^a, 8.30, 4.94, 2.80)$	$(\infty, \infty, 15, 2, 4)$	0.001
$C_{1,1}$	$(\infty, \infty, 13.39, 12.63, 4.01)$	$(\infty, \infty, 14, 14, 5)$	0.015
$C_{1,2}$	$(\infty, \infty, 13.45, 13.03, 4.06)$	$(\infty, \infty, 14, 14, 5)$	0.014
$C_{2,0}$	$(\infty, -^b, 9.19, 3.98, 2.74)$	$(\infty, \infty, 26, 14, 6)$	0.001
$C_{2,1}$	$(\infty, \infty, 24.32, 12.71, 4.89)$	$(\infty, \infty, 26, 14, 6)$	0.021
$C_{2,2}$	$(\infty, \infty, 24.71, 13.20, 5.08)$	$(\infty, \infty, 26, 14, 6)$	0.048

<sup>a</sup>46% codes of this code set are free of 4-cycles.

<sup>b</sup>13% codes of this code set are free of 4-cycles.

3. From Fig. 3, it can be seen that each code designed with the metric (2) outperforms all that codes designed with the metric (1) in the high SNR region between 2.6dB and 2.8dB. In addition, among the four codes designed with the metric (2) (i.e.,  $B_1$ – $B_4$ ),  $B_4$  with the best ACE spectrum achieves the lowest BER, and  $B_1$  with the poorest ACE spectrum holds the highest BER. In such case, the simulation results in Fig. 3 are consistent with that increasing the ACE spectrum is expected to have a positive effect on the error performance [11]–[15].

*Example 4:* This example considers the design of the irregular (1008, 504) QC-LDPC codes with the one-dimension circulant-size  $N = 36$  and the VN-degree distribution  $\gamma(x) = 0.46429x^2 + 0.28571x^3 + 0.03571x^4 + 0.10714x^5 + 0.10714x^{15}$  using the MM-QC-PEGA. Besides,  $N$  cannot be too large, or the VN-degree distribution will change heavily from the DE-optimized one used in Example 3. For convenience, denote  $C_{\alpha,\beta}$ ,  $\alpha = 1, 2$ ,  $\beta = 0, 1, 2$  as the code (or code set) designed by the QC-PEGA [33] with the metric ( $\alpha$ ) when  $\beta = 0$ , and as the code (or code set) designed by the  $\beta$ -edge M-QC-PEGA with the metric ( $\alpha$ ) when  $\beta = 1$  or 2. However, the CP-PEGA [34], which requires each nonzero circulant of the check matrix to be a CPM, does not suit for this example.

Exact 1000 QC-LDPC codes of each code set are randomly constructed. In addition, the average ACE spectrum, the maximum ACE spectrum, and the frequency associated with the maximum ACE spectrum of each code set are presented in Table IV. The average ACE spectra in Table IV show that the QC-PEGA [33] sometimes results in 4-cycles while the MM-QC-PEGA avoid generating 4-cycles effectively. In addition, the MM-QC-PEGA overwhelmingly outperforms the QC-PEGA [33] in terms of the average ACE spectra, indicating that the MM-

QC-PEGA is much more effective than the QC-PEGA [33] for designing the QC-LDPC code with larger girth as well as better ACE spectrum. However, one code in the code set  $C_{1,0}$  achieves the largest ACE spectrum  $(\infty, \infty, 15, 2, 4)$  among the code sets  $C_{1,\beta}$ ,  $\beta = 0, 1, 2$ . But this code may hold a high error floor since there apparently contains some small ETSs, such as the  $(4, 2)$  ETS and the  $(5, 4)$  ETS, each of which consists of a single cycle. Instead, the second largest ACE spectrum of the code set  $C_{1,0}$ , i.e.,  $(\infty, \infty, 14, 14, 5)$ , which equals the maximum ACE spectra of  $C_{1,\beta}$ ,  $\beta = 1, 2$  but occurs in a much smaller frequency 0.001, should be considered as a better one. Furthermore, for the codes designed by the MM-QC-PEGA, their ACE spectra under the metric (2) greatly surpass that under the metric (1), which is also consistent with the simulation results in Example 3 and that in [14]. Meanwhile, under the same metric, the average ACE spectrum achieved by the two-edge M-QC-PEGA is slightly better than that achieved by the one-edge M-PEGA. When compare Table IV to Table II, the average and the maximum ACE spectra achieved by the MM-QC-PEGA in Table IV are better than that achieved by the MM-PEGA in Table II.<sup>6</sup> In such case, with regard to designing the irregular LDPC codes with better ACE spectra, the QC-LDPC codes, which have proper circulant-sizes and are designed by the MM-QC-PEGA, may be more preferable than the non-QC-LDPC codes designed by the MM-PEGA.

To perform the BER estimations, one of the codes whose ACE spectra are the maximum among the first 100 constructed codes of each code set is randomly selected.<sup>7</sup> The ACE spectrum of  $C_{1,0}$  is  $(\infty, \infty, 14, 14, 3)$ , and that of  $C_{1,2}$  is  $(\infty, \infty, 14, 14, 4)$ , and that of the other codes remain the same as the maximum ACE spectra of their own code sets in Table IV. The BER performances of these selected codes are presented in Fig. 4, where the BER performances of these codes match their ACE spectra well, i.e. the code with a better ACE spectrum achieves a lower BER. Furthermore, both  $C_{2,1}$  and  $C_{2,2}$  slightly outperform the best code in Fig. 3 (i.e.,  $B_4$ ) with regard

<sup>6</sup>As the VN-degree distributions used in Example 3 and Example 4 only differ a little from each other, the comparison between Table II (Example 3) and Table IV (Example 4) is quite fair. Also, to verify this, under the VN-degree distribution used in Example 4, another 1000 codes have been randomly constructed using each of the MM-PEGA with edge-trials 1 and 2 and with metrics (1) and (2). Then, the statistical results of these codes show that both their average and maximum ACE spectra are inferior to the corresponding ones of Table II.

<sup>7</sup>This is because the error estimations in this example have been performed after 100 codes of each code set are constructed. However, to make the aforementioned investigation of the ACE spectra more accurate, another 900 codes of each code set have been constructed. Furthermore, in the rest of this paper, the codes for performing error estimations are always selected from 100 candidate codes.

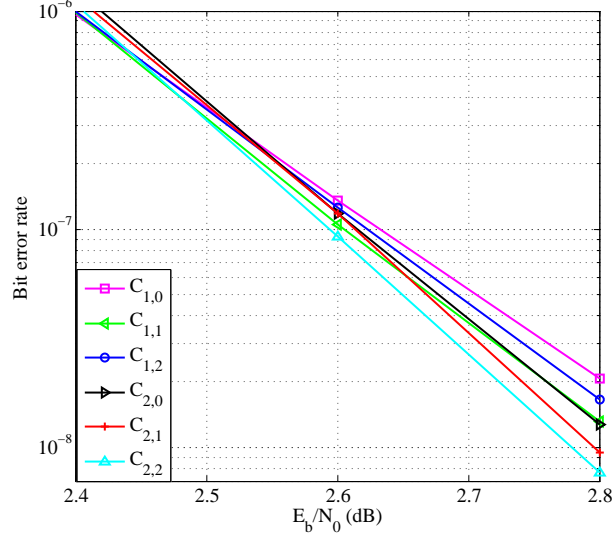


Fig. 4. BER performances of the irregular (1008, 504) QC-LDPC codes with  $N = 36$  designed in Example 4.

to the BER at  $E_b/N_0 = 2.8\text{dB}$ . On the one hand, this is interesting since  $C_{2,1}$  and  $C_{2,2}$  are QC while  $B_4$  is non-QC. On the other hand, it is reasonable since  $C_{2,1}$  and  $C_{2,2}$  hold a better ACE spectrum compared to that of  $B_4$ . In summary, the simulation results in this example are also consistent with that better ACE spectra generally result in lower error rates [11]–[15].

### B. Regular LDPC Codes

*Example 5:* This example considers the design of the regular (1024, 512) QC-LDPC codes with VN-degree-three. For each valid one-dimension circulant-size  $N$  from  $\{2^0, 2^1, \dots, 2^9\}$  and each design algorithm of the QC-PEGA [33], the CP-PEGA [34], the one-edge M-QC-PEGA, and the two-edge M-QC-PEGA, exact 100 QC-LDPC codes are randomly constructed.<sup>8</sup> For each code set of 100 codes with the same  $N$  and the same design algorithm, their average and minimum VNLGDs are presented in Table V. At the same time, the frequency that the minimum VNLGD occurs among all the VNLGDs of the codes from a same code set is given in Table V too. For each specific  $N$  from Table V(a) to Table V(d), both the average and the minimum VNLGDs keep nonincreasing while the frequencies keep nondecreasing,<sup>9</sup> indicating that the MM-QC-PEGA, especially that with a larger edge-trials, is more effective than the QC-PEGA [33] and the CP-PEGA [34] for designing the QC-LDPC codes with better VNLGDs.

<sup>8</sup>When constructing regular LDPC codes, it makes no difference to use metric (1) or metric (2).

<sup>9</sup> $N = 2^4$  is the only one exception where the frequencies do not keep nondecreasing.



TABLE V  
VNLGDs OF DIFFERENT QC-LDPC CODES IN EXAMPLE 5

(a) QC-PEGA [33]

$N$	Average	Minimum	Freq
$2^0$	$0.1058x^4 + 0.8942x^6$	$0.0293x^8 + 0.9707x^{10}$	0.01
$2^1$	$0.0078x^4 + 0.1628x^8 + 0.8294x^{10}$	$0.1426x^8 + 0.8574x^{10}$	0.01
$2^2$	$0.0080x^4 + 0.0004x^6 + 0.1619x^8 + 0.8297x^{10}$	$0.1250x^8 + 0.8750x^{10}$	0.01
$2^3$	$0.0092x^4 + 0.0018x^6 + 0.1727x^8 + 0.8163x^{10}$	$0.0625x^8 + 0.9375x^{10}$	0.01
$2^4$	$0.0095x^4 + 0.0053x^6 + 0.1914x^8 + 0.7938x^{10}$	$0.0312x^8 + 0.9688x^{10}$	0.04
$2^5$	$0.0160x^4 + 0.0096x^6 + 0.2078x^8 + 0.7666x^{10}$	$1.0000x^{10}$	0.02
$2^6$	$0.0325x^4 + 0.0619x^6 + 0.2694x^8 + 0.6362x^{10}$	$1.0000x^{10}$	0.01
$2^7$	$0.1500x^4 + 0.1525x^6 + 0.3500x^8 + 0.3475x^{10}$	$1.0000x^{10}$	0.03
$2^8$	$0.3825x^4 + 0.1625x^8 + 0.4550x^8$	$1.0000x^8$	0.22
$2^9$	$0.4500x^4 + 0.5500x^6$	$1.0000x^6$	0.38

(b) CP-PEGA [34]

$N$	Average	Minimum	Freq
$2^0$	$0.1058x^8 + 0.8942x^{10}$	$0.0293x^8 + 0.9707x^{10}$	0.01
$2^1$	$0.1748x^8 + 0.8252x^{10}$	$0.0566x^8 + 0.9434x^{10}$	0.01
$2^2$	$0.1732x^8 + 0.8268x^{10}$	$0.0312x^8 + 0.9688x^{10}$	0.01
$2^3$	$0.1704x^8 + 0.8296x^{10}$	$0.0312x^8 + 0.9688x^{10}$	0.01
$2^4$	$0.2133x^8 + 0.7867x^{10}$	$1.0000x^{10}$	0.02
$2^5$	$0.2622x^8 + 0.7378x^{10}$	$1.0000x^{10}$	0.09
$2^6$	$0.4044x^8 + 0.5956x^{10}$	$1.0000x^{10}$	0.12
$2^7$	$0.5900x^8 + 0.4100x^{10}$	$1.0000x^{10}$	0.06

(c) One-Edge M-QC-PEGA

$N$	Average	Minimum	Freq
$2^0$	$0.1058x^8 + 0.8942x^{10}$	$0.0293x^8 + 0.9707x^{10}$	0.01
$2^1$	$0.1303x^8 + 0.8697x^{10}$	$0.0195x^8 + 0.9805x^{10}$	0.01
$2^2$	$0.1196x^8 + 0.8804x^{10}$	$1.0000x^{10}$	0.03
$2^3$	$0.1375x^8 + 0.8625x^{10}$	$1.0000x^{10}$	0.15
$2^4$	$0.1620x^8 + 0.8380x^{10}$	$1.0000x^{10}$	0.32
$2^5$	$0.1191x^8 + 0.8809x^{10}$	$1.0000x^{10}$	0.63
$2^6$	$0.1731x^8 + 0.8269x^{10}$	$1.0000x^{10}$	0.70
$2^7$	$0.1225x^8 + 0.8775x^{10}$	$1.0000x^{10}$	0.85
$2^8$	$1.0000x^8$	$1.0000x^8$	1.00
$2^9$	$1.0000x^6$	$1.0000x^6$	1.00

(d) Two-Edge M-QC-PEGA

$N$	Average	Minimum	Freq
$2^0$	$0.0004x^8 + 0.9996x^{10}$	$1.0000x^{10}$	0.97
$2^1$	$0.0003x^8 + 0.9997x^{10}$	$1.0000x^{10}$	0.99
$2^2$	$0.0006x^8 + 0.9994x^{10}$	$1.0000x^{10}$	0.99
$2^3$	$0.0017x^8 + 0.9983x^{10}$	$1.0000x^{10}$	0.98
$2^4$	$0.0017x^8 + 0.9983x^{10}$	$1.0000x^{10}$	0.99
$2^5$	$1.0000x^{10}$	$1.0000x^{10}$	1.00
$2^6$	$1.0000x^{10}$	$1.0000x^{10}$	1.00
$2^7$	$1.0000x^{10}$	$1.0000x^{10}$	1.00
$2^8$	$1.0000x^8$	$1.0000x^8$	1.00
$2^9$	$1.0000x^6$	$1.0000x^6$	1.00

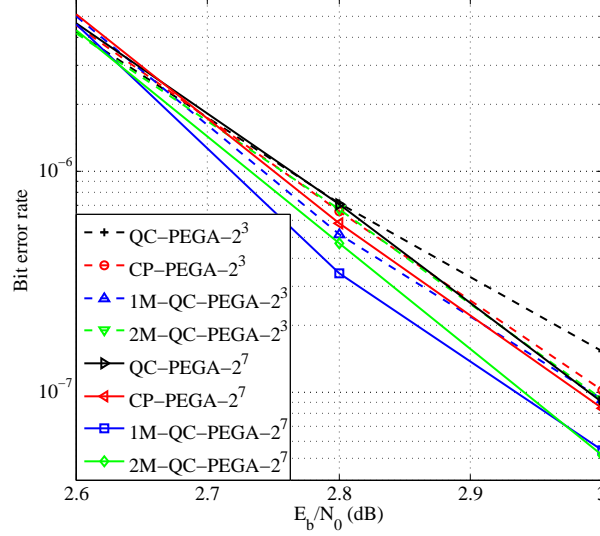


Fig. 5. BER performances of the regular (1024, 512) QC-LDPC codes with  $N = 2^3$  and  $N = 2^7$  designed in Example 5.

Furthermore, according to the comparison between the QC-LDPC codes with the different  $N$ s in the same table from Table V(a) to Table V(d), the codes with moderate  $N$ s, such as  $N = 2^5$ ,  $2^6$ ,  $2^7$ , have larger chances to achieve the optimal VNLGDs. By the way, if  $N$  is too large, such as  $N = 2^8$ ,  $2^9$ , short inevitable cycles will form in the QC-LDPC code with the assumed  $N$ , which also coincides with the results in [27].

For  $N = 2^3$  and  $N = 2^7$  in each table from Table V(a) to Table V(d), one of the codes whose VNLGDs are the minimum is randomly selected to perform the BER estimations. The error performances of the selected codes are presented in Fig. 5, where the codes are labelled based on their design algorithms and their one-dimension circulant-sizes, such as 1M-QC-PEGA- $2^3$  denotes the code which is designed by the one-edge M-QC-PEGA with  $N = 2^3$ . From Fig. 5, it can be seen that the codes designed by the MM-QC-PEGA outperform that codes designed by the QC-PEGA [33] and the CP-PEGA [34] at most 0.1dB in the high SNR region between 2.8dB and 3.0dB. At the same time, the code 2M-QC-PEGA- $2^7$  slightly outperforms the regular (1024, 512) non-QC-LDPC codes designed by the MM-PEGA in Example 6 at  $E_b/N_0 = 3.0$ dB, presenting another sample for that the QC-LDPC codes designed by the MM-QC-PEGA can sometimes outperform the non-QC-LDPC codes designed by the MM-PEGA.

*Example 6:* This example considers the design of three different types of regular non-QC-LDPC codes with VN-degree-three using the MM-PEGA. These types of codes are the rate-1/2 (1024, 512), the rate-2/3 (1536, 1024), and the rate-3/4 (2048, 1536) LDPC codes. For each

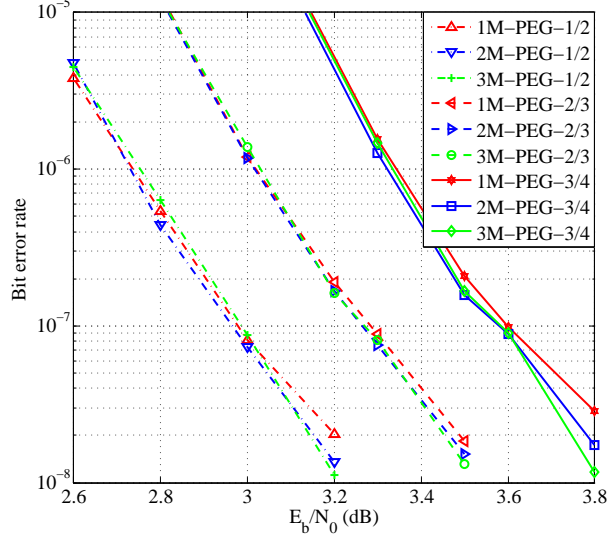


Fig. 6. BER performances of a sequence of codes with rates 1/2, 2/3, and 3/4 designed in Example 6.

rate and each of the MM-PEGA with edge-trials 1, 2, and 3, exact 100 codes are randomly constructed. For convenience, different codes (code sets) are denoted based on their design algorithms and rates, such as 1M-PEGA-1/2 denotes the code (code set) which is designed by the one-edge M-PEGA and has the rate 1/2. Furthermore, one of the codes whose VNLGDs are the minimum is randomly selected from each code set to perform the BER estimations,<sup>10</sup> where the VNLGD of 1M-PEGA-1/2 is  $0.0293x^8 + 0.9707x^{10}$ , and that of the other rate-1/2 codes are  $x^{10}$ , and that of the left codes are  $x^8$ . At the same time, the error performances of the selected codes are presented in Fig. 6. Furthermore, according to the comparisons among the three codes with the same rates, it's verified that increasing the edge-trials of the MM-PEGA is expected to have a positive effect on the VNLGDs as well as the error performances of the codes designed by the MM-PEGA.

## VI. CONCLUSIONS

In this paper, the metric (1) and the metric (2) were formulated firstly so that the PEG algorithm [10] and the ACE constrained PEG algorithm [14] were unified into one integrated algorithm, i.e., the M-PEGA. Then, as an improvement for the M-PEGA, the MM-PEGA was proposed, which is implemented under the framework of the M-PEGA but adopts Strategy 2 instead of

<sup>10</sup>The codes 1M-PEGA-1/2 and 2M-PEGA-1/2 are selected from the code sets of Example 5, which are designed by the MM-QC-PEGA and have one-dimension circulant-size 1.

Strategy 1 to select the CNs. It was illustrated that the one-edge M-PEGA is equivalent to the M-PEGA. In addition, to calculate the multi-edge local girths used in Strategy 2, a DFS like algorithm was proposed. It was verified by the simulation results in Example 3 and Example 6 that increasing the edge-trials of the MM-PEGA is expected to have a positive effect on the cycle-structure as well as the error performance of the non-QC LDPC code designed by the MM-PEGA. Furthermore, to the best of our knowledge, the maximum ACE spectra reported in Table II are the best among that spectra of the codes designed under the similar design parameters in the literature.

Meanwhile, the MM-QC-PEGA was proposed, which is implemented under the framework of the QC-PEGA [33] but adopts a different selection strategy. On the one hand, the MM-QC-PEGA can construct the QC-LDPC codes with any predefined valid design parameters. On the other hand, the undetectable cycles in the QC-LDPC codes designed by the QC-PEGA [33] and the CP-PEGA [34] become detectable and even avoidable in the codes designed by the MM-QC-PEGA. It was illustrated that the  $r$ -edge M-PEGA is equivalent to the  $r$ -edge MM-QC-PEGA when the one-dimension circulant-size  $N = 1$ . It was verified by the simulation results in Example 4 and Example 5 that: 1) Compared to the QC-PEGA [33] and the CP-PEGA [34], the MM-QC-PEGA is expected to achieve better cycle-structures as well as better error performances. 2) Increasing the edge-trials of the MM-QC-PEGA is expected to have a positive effect on the cycle-structure as well as the error performance of the QC-LDPC code designed by the MM-QC-PEGA.

When designing the LDPC code with the given length, rate, and VN-degree distribution, the QC-LDPC code, which has a proper one-dimension circulant-size and is designed by the MM-QC-PEGA, may be more preferable than the non-QC-LDPC code designed by the MM-PEGA, with regard to the following three observations on the simulation results: 1) The average and the maximum ACE spectra of the QC-LDPC codes designed by the MM-QC-PEGA reported in Table IV are even better than the corresponding ones of the non-QC-LDPC codes designed by the MM-PEGA in Table II. 2) When the one-dimension circulant-size  $N$  of the QC-LDPC codes designed by the MM-QC-PEGA in Table V(c) and Table V(d) increase, a better average and a better minimum VNLGD may be achieved. 3) Some QC-LDPC codes designed by the MM-QC-PEGA outperform the non-QC LDPC codes designed by the MM-PEGA in terms of the BER in the high SNR region, such as the best QC-LDPC code  $C_{2,2}$  in Example 4 slightly outperforms the best non-QC-LDPC code  $B_4$  in Example 3, and the best QC-LDPC code 2M-QC-PEGA-2<sup>7</sup> in Example 5 slightly outperforms the best rate-1/2 non-QC-LDPC code 3M-PEGA-1/2 in

Example 6. But instead, the computational complexity of the  $r$ -edge M-QC-PEGA is higher than that of the  $r$ -edge M-PEGA.

## REFERENCES

- [1] X. He, L. Zhou, J. Du, and Z. Shi, “The multi-step PEG and ACE constrained PEG algorithms can design the LDPC codes with better cycle-connectivity,” in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2015.
- [2] R. G. Gallager, “Low-density parity-check codes,” *IRE Trans. Inf. Theory*, vol. IT-8, no. 1, pp. 21–28, Jan. 1962.
- [3] T. Richardson, “Error floor of LDPC codes,” in *Proc. 41st Annu. Allerton Conf. Commun., Control, Comput.*, Oct. 2003.
- [4] A. McGregor and O. Milenkovic, “On the hardness of approximating stopping and trapping sets,” *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1640–1650, Apr. 2010.
- [5] R. M. Tanner, “A recursive approach to low complexity codes,” *IEEE Trans. Inf. Theory*, vol. IT-27, no. 5, pp. 533–547, Sep. 1981.
- [6] X. Zheng, F. C. M. Lau, and C. K. Tse, “Constructing short-length irregular LDPC codes with low error floor,” *IEEE Trans. Commun.*, vol. 58, no. 10, pp. 2823–2834, Oct. 2010.
- [7] R. Asvadi, A. H. Banihashemi, and M. Ahmadian-Attari, “Lowering the error floor of LDPC codes using cyclic liftings,” *IEEE Trans. Inf. Theory*, vol. 57, no. 4, pp. 2213–2224, Apr. 2011.
- [8] S. Khazraie, R. Asvadi, and A. H. Banihashemi, “A PEG construction of finite-length LDPC codes with low error floor,” *IEEE Commun. Lett.*, vol. 16, no. 8, pp. 1288–1291, Aug. 2012.
- [9] M. Diouf, D. Declercq, S. Ouya, and B. Vasic, “A PEG-like LDPC code design avoiding short trapping sets,” in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2015.
- [10] X.-Y. Hu, E. Eleftheriou, and D. M. Arnold, “Regular and irregular progressive edge-growth Tanner graphs,” *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.
- [11] T. Tian, C. R. Jones, J. D. Villasenor, and R. D. Wesel, “Selective avoidance of cycles in irregular LDPC code construction,” *IEEE Trans. Commun.*, vol. 52, no. 8, pp. 1242–1247, Aug. 2004.
- [12] H. Xiao and A. H. Banihashemi, “Improved progressive-edge-growth (PEG) construction of irregular LDPC codes,” *IEEE Commun. Lett.*, vol. 8, no. 12, pp. 715–717, Dec. 2004.
- [13] D. Vukobratović, A. Djurendić, and V. Šenk, “ACE spectrum of LDPC codes and generalized ACE design,” in *Proc. IEEE Int. Conf. Commun.*, Jun. 2007.
- [14] D. Vukobratović and V. Šenk, “Generalized ACE constrained progressive edge-growth LDPC code design,” *IEEE Commun. Lett.*, vol. 12, no. 1, pp. 32–34, Jan. 2008.
- [15] —, “Evaluation and design of irregular LDPC codes using ACE spectrum,” *IEEE Trans. Commun.*, vol. 57, no. 8, pp. 2272–2278, Aug. 2009.
- [16] R. Asvadi, A. H. Banihashemi, and M. Ahmadian-Attari, “Design of finite-length irregular protograph codes with low error floors over the binary-input AWGN channel using cyclic liftings,” *IEEE Trans. Commun.*, vol. 60, no. 4, pp. 902–907, Apr. 2012.
- [17] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms (second edition)*. Cambridge, MA, USA: MIT Press, 2001.
- [18] Z. Li, L. Chen, L. Zeng, S. Lin, and W. Fong, “Efficient encoding of quasi-cyclic low-density parity-check codes,” *IEEE Trans. Commun.*, vol. 54, no. 1, pp. 71–81, Jan. 2006.
- [19] K. Liu, S. Lin, and K. Abdel-Ghaffar, “A revolving iterative algorithm for decoding algebraic cyclic and quasi-cyclic LDPC codes,” *IEEE Trans. Commun.*, vol. 61, no. 12, pp. 4816–4827, Dec. 2013.

- [20] J. Li, K. Liu, S. Lin, and K. Abdel-Ghaffar, "Algebraic quasi-cyclic LDPC codes: construction, low error-floor, large girth and a reduced-complexity decoding scheme," *IEEE Trans. Commun.*, vol. 62, no. 8, pp. 2626–2637, Aug. 2014.
- [21] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low-density parity-check codes based on finite geometries: a rediscovery and new results," *IEEE Trans. Inf. Theory*, vol. 47, no. 7, pp. 2711–2736, Nov. 2001.
- [22] L. Lan, L. Zeng, Y. Y. Tai, L. Chen, S. Lin, and K. Abdel-Ghaffar, "Construction of quasi-cyclic LDPC codes for AWGN and binary erasure channels: a finite field approach," *IEEE Trans. Inf. Theory*, vol. 53, no. 7, pp. 2429–2458, Jul. 2007.
- [23] S. Myung and K. Yang, "A combining method of quasi-cyclic LDPC codes by the Chinese Remainder Theorem," *IEEE Commun. Lett.*, vol. 9, no. 9, pp. 823–825, Sep. 2005.
- [24] X. Jiang and M. H. Lee, "Large girth quasi-cyclic LDPC codes based on the Chinese Remainder Theorem," *IEEE Commun. Lett.*, vol. 13, no. 5, pp. 342–344, May 2009.
- [25] X. Jiang, X.-G. Xia, and M. H. Lee, "Efficient progressive edge-growth algorithm based on Chinese Remainder Theorem," *IEEE Trans. Commun.*, vol. 62, no. 2, pp. 442–451, Feb. 2014.
- [26] J. Huang, L. Liu, W. Zhou, and S. Zhou, "Large-girth nonbinary QC-LDPC codes of various lengths," *IEEE Trans. Commun.*, vol. 58, no. 12, pp. 3436–3447, Dec. 2010.
- [27] H. Park, S. Hong, J.-S. No, and D.-J. Shin, "Design of multiple-edge protographs for QC LDPC codes avoiding short inevitable cycles," *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4598–4614, Jul. 2013.
- [28] D. G. M. Mitchell, R. Smarandache, and D. J. Costello, Jr., "Quasi-cyclic LDPC codes based on pre-lifted protographs," *IEEE Trans. Inf. Theory*, vol. 60, no. 10, pp. 5856–5874, Oct. 2014.
- [29] M. O'Sullivan, "Algebraic construction of sparse matrices with large girth," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 718–727, Feb. 2006.
- [30] I. E. Bocharova, F. Hug, R. Johannesson, B. D. Kudryashov, and R. V. Satyukov, "Searching for voltage graph-based LDPC tailbiting codes with large girth," *IEEE Trans. Inf. Theory*, vol. 58, no. 4, pp. 2265–2279, Apr. 2012.
- [31] M. Karimi and A. Banihashemi, "On the girth of quasi-cyclic protograph LDPC codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4542–4552, Jun. 2013.
- [32] M. P. C. Fossorier, "Quasi-cyclic low-density parity-check codes from circulant permutation matrices," *IEEE Trans. Inf. Theory*, vol. 50, no. 8, pp. 1788–1793, Aug. 2004.
- [33] Z. Li and B. V. K. V. Kumar, "A class of good quasi-cyclic low-density parity check codes based on progressive edge growth graph," in *Proc. 38th Asilomar Conf. Signals, Syst. Comput.*, Nov. 2004.
- [34] Y.-K. Lin, C.-L. Chen, Y.-C. Liao, and H.-C. Chang, "Structured LDPC codes with low error floor based on PEG Tanner graphs," in *Proc. IEEE Int. Symp. on Circuits and Systems*, May 2008.
- [35] J. Xu, L. Chen, I. Djurdjevic, S. Lin, and K. Abdel-Ghaffar, "Construction of regular and irregular LDPC codes: geometry decomposition and masking," *IEEE Trans. Inf. Theory*, vol. 53, no. 1, pp. 121–134, Jan. 2007.
- [36] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.
- [37] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.